UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/731,515	12/09/2003	Marcin Sawicki	60001.295US01	3732

27488	7590	06/26/2006	EXAMINER

MERCHANT & GOULD (MICROSOFT)
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

BOTTS, MICHAEL K

ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 06/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/731,515	SAWICKI ET AL.
	Examiner	Art Unit	
	Michael K. Botts	2176	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _29 March 2006_.

2a)☒ This action is **FINAL.**   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-23_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-23_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _09 December 2003 and 29 March 2006_ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1.   This document is a Final Office Action on the merits.   This action is responsive

to the following communications: Amendment, which was filed on March 29, 2006.

2.   Claims 1-23 are currently pending in the case, with claims 1, 11 and 19 being the

independent claims. Claims 1, 2, 11, and 19 have been amended.

3.   The Abstract is objected to.

4.   The specification is objected to.

5.   Claim 2 was objected to. Applicants have appropriately amended the claim to

obviate the objection as suggested by the Examiner. Accordingly, the objection to claim

2 is withdrawn.

6.   Claims 3, 4, 14, 15, 21, and 22 are objected to.

7.   Claims 1-23 are rejected.

Information Disclosure Statement

8.   Applicants filed a document designated as an Information Disclosure Statement

on May 7, 2006. The document filed is not in the form of an information disclosure

statement, and does not provide sufficient information for the Examiner to review and

consider the information provided. The document presents factual evidence relating to

the patentability of the invention without proper affidavit support. Accordingly, the

document is acknowledge as having been received, but has not been considered by the

Examiner.

Abstract of the Disclosure

The abstract of the disclosure is objected to because of the use it does not accurately reflect the invention claimed. The statement that the invention "may be manipulated on a server or anywhere even when the application creating the ML document is not present" is not claimed, and is essentially inherent in the markup language itself. In addition, the statement that the invention fields "may be manipulated when the ML document is parsed by other applications," similarly identifies a property of a markup language, rather than that of the invention itself. Finally, the statement that "field definition information (i.e. properties) are save in a markup language (ML) document without data loss, while allowing the filed structures to be parsed by ML-aware applications and to be read by ML programmers" also merely states inherent properties of the markup language, rather than stating a concise description of the invention. Correction is required. See MPEP § 608.01(b).

Drawings

9.   The drawings are objected to because it is unclear which drawing elements are being identified by the reference characters and lead lines from the following reference characters: 310, 320, 330, 410, 420, 430, 440, and 450. It is suggested that the item being identified be cited specifically in the specification, rather than an arrow or lead line to a line of complex computer code with only a general description in the specification.

10.   The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4)

because reference character "440" has been used to designate two different drawing

elements. It is suggested that the Applicant designate the elements "440A" and "440B"

or something similar to indicate that the elements are of the same type, namely

instructions, but they are not the same instruction.

11.   The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4)

because reference character "410" has been used to designate both "instruction 410"

and fldChar 410. See, disclosure, page 9, lines 17, and 19.

12.   In response to the above identified objections, corrected drawing sheets in

compliance with 37 CFR 1.121(d), or amendment to the specification to add the

reference character(s) in the description in compliance with 37 CFR 1.121(b) are

required in reply to the Office action to avoid abandonment of the application. Any

amended replacement drawing sheet should include all of the figures appearing on the

immediate prior version of the sheet, even if only one figure is being amended. Each

drawing sheet submitted after the filing date of an application must be labeled in the top

margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If

the changes are not accepted by the examiner, the applicant will be notified and

informed of any required corrective action in the next Office action. The objection to the

drawings will not be held in abeyance.

The Specification

13.   Applicant is required to update the status (pending, allowed, etc.) of all parent

priority applications in the first line of the specification. The status of all citations of U.S.

filed applications in the specification should also be updated where appropriate.

14.   Applicants submitted a compact disc containing the computer code previously

submitted in the written application. Applicants have amended the specification to

reflect that the code is contained on the compact disc. However, the code on the

compact disc submitted by the Applicants does not conform to the standards set forth in

37 CFR 1.96(c),as required in the Non-Final Office Action. Specifically, the data

submitted on the compact disc is not in American Standard Code for Information

Interchange (ASCII) format. See, 37 CFR 1.96(c)(3)(i).

        Applicant is required to file a computer program listing appendix on compact disc

in compliance with 37 CFR 1.96(c).

Claims Objections

15.   **Claims 3, 4, 14, 15, 21, and 22** are objected to because of the following

informalities: The term "richly formatted text" is not expressly defined in the application.

From the context of the use of the term in disclosure and in the claims, it is believed that

applicants intended to refer to Microsoft Corporation's Rich Text Format (RTF)

standard, and the claims will be read that "richly formatted text" is the equivalent of Rich

Text Format (RTF) for the remainder of this Office Action. Appropriate correction is

required.

Claims Rejections – 35 U.S.C. 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

16.   **Claims 1 and 7-10** are rejected under 35 U.S.C. 102(b) as being clearly

anticipated by Ayers, I., "AbiWord's Potential," Linux Gazette, Issue 43, July 1999, last

downloaded by the Examiner on December 20, 2005, from:

www.linuxgazette.com/issue43/ayers.html, downloaded pages 1-4, [hereinafter "Ayers"].

Regarding **independent claim 1, as amended**, Ayers teaches:

A method for representing field structures in a markup language

document, comprising:

inputting an application document that has been generated by an

application that uses a file format that is specific to the application;

(The specification discloses that the invention "provides a method to represent an

application's native field structures in markup language (ML) such as XML." See,

Disclosure, page 6, lines 12-13. Ayers teaches: "The most significant difference

between AbiWord and nearly every other word processor available is the nature of the

native file format. An *.abw file is written in XML and thus is also in ASCII format; the

files can be read by any text editor." See, Ayers, page 2, fourth paragraph. Therefore,

Ayers teaches the limitation of "a document that has been generated by an application

that uses a file format that is specific to the application," and more specifically, teaches

a native file format in a markup language, specifically XML.)

> *determining properties corresponding to a field that relates to at least one*
>
> *section of an application document;*

(It is noted that the term "properties" is not specifically defined in the specification. The

disclosure associates the term in a general sense with simple and complex fields, as in

the following: "the properties of the complex field (when the field is a complex field) are

mapped into elements, attributes, and values of the ML file." Disclosure, page 18, lines

9-10. "Attributes" are also noted as one of the properties elsewhere in the disclosure.

See, disclosure, page 3, line 26. The disclosure also defines that an element may have

no attribute associated with it. See, disclosure, page 3, lines 27-28. Further evidence

that the term refers to the general appearance of the document is taken from the

disclosure, stating: "the fields and the properties associated with the fields may change

from page to page, section to section, chapter to chapter and the like." Disclosure,

page 18, lines 22-23. Based on the above analysis, it is believed that the applicants

intended the term "properties" to be used in the general sense of the appearance of the

text, and such definition will be used for the remainder of this Office Action. In support

of the above definition of "properties" as known to one of ordinary skill in the art at the

time of the invention, see, Harold, Elliotte Rusty, "XML Bible," IDG Books Worldwide,

1999, pages 369-388.)

> *mapping the properties of the field into at least one of a markup language*
>
> *element, an attribute, and a value; and*

(See, Ayers, page 3, last two full paragraphs and citing code example from the

screenshot and showing font and line properties and values. Note that Ayers teaches

that the code is the markup language mapped from the properties of the document field

displayed in the screenshot.)

> *storing the properties of the field in the markup language document,*
>
> *whereby applications different from the application can understand the mapped*
>
> *list properties stored in the markup language document..*

(See, Ayers, page 3, last two full paragraphs and citing code example from the

screenshot. Note that Ayers teaches saving the document shown in the screenshot,

which creates the saved code that is saved in memory.

Ayers also teaches: "The most significant difference between AbiWord and

nearly every other word processor available is the nature of the native file format. An

*.abw file is written in XML and thus is also in ASCII format; **the files can be read by**

any text editor." See, Ayers, page 2, fourth paragraph, emphasis added.)

Regarding **dependent claim 7**, Ayers teaches:

> *The method of Claim 1, further comprising:*

> *determining properties corresponding to an additional field that relates to*
>
> *at least one section of the application document;*
>
> *mapping the properties of the additional field into at least one of a markup*
>
> *language element, an attribute, and a value; and*
>
> *storing the properties of the additional field in the markup language*
>
> *document.*

(It is noted that this claim is read as the method including to edit and add or insert text to

a document and then to convert to XML and save the resulting combined document.

See, Ayers, page 3, last two lines, teaching the editing, or modification, of an

XML document in AbiWord.)

Regarding **dependent claim 8**, claim 8 incorporates substantially similar subject matter

as claimed in claim 1, and in further view of the following, is rejected along the same

rationale.

It is noted that claim 1 specifies storing "the field," which is a singular "field."

See, claim 1, lines, 5 and 9. There may be many fields in a document. Claim 8 further

limits claim 1 by specifying a repetition of the field storage process until "all fields" have

been stored. Claim 8 merely specifies completing the field storage process, which is

the equivalent of storing the document.

Ayers teaches the storage of the document, which teaches the storage of all

fields in the document. See, Ayers, page 3, last two full paragraphs and citing code

example from the screenshot. Note that Ayers teaches saving the document shown in

the screenshot, which creates the saved code that is saved in memory.

Regarding **dependent claim 9**, Ayers teaches:

> The method of Claim 1, wherein the properties of the fields stored in the
>
> markup language document are understood by an application that understands
>
> the markup language when the field is not native to the application.

(See, Ayers, page 2, third full paragraph, teaching that an AbiWord document is saved

in an *.abw file written in XML and that the files can be read by any text editor. IN

addition, Ayers teaches fields stored in XML that are understood by an application that

understands the markup language when the field is not native to the application. See,

Ayers, page 2, fourth paragraph teaching, as follows: "AbiWord can also save in the

HTML and RTF formats, both of which are accessible with word processors such as

MS-Word and WordPerfect." Therefore, AbiWord's fields in XML are understood by

markup languages not native to AbiWord, such as HTML.)

Regarding **dependent claim 10**, Ayers teaches:

> The method of Claim 1, wherein the markup language document is
>
> manipulated on a server to substantially reproduce the field of the application
>
> document notwithstanding the presence of an application that generated the
>
> markup language document.

(See, Ayers, page 2, third full paragraph, teaching that an AbiWord document is saved

in an *.abw file written in XML and that the files can be read by any text editor. See

also, Ayers, page 3, bottom two lines, teaching that the document save in AbiWord may

be modified by hand in an editor rather than a word processor. It is inherent from the

ability to be read by any text editor and be modified by hand that the document may be

manipulated on any suitable computing device, including a server.)

17.   It is noted that any citations to specific, pages, columns, lines, or figures in the

prior art references and any interpretation of the references should not be considered to

be limiting in any way. A reference is relevant for all it contains and may be relied upon

for all that it would have reasonably suggested to one having ordinary skill in the art.

See, MPEP 2123.

Claims Rejection – 35 U.S.C. 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

18.   **Claims 2-6, and 11-23** are rejected under 35 U.S.C. 103(a) as being

unpatentable over Ayers, I., "AbiWord's Potential," Linux Gazette, Issue 43, July 1999,

last downloaded by the Examiner on December 20, 2005, from:

www.linuxgazette.com/issue43/ayers.html, [hereinafter "Ayers"], in view of W3C, "XML

Schema Part 0: Primer, W3C Recommendation, 2 May 2001," last downloaded by the

Examiner on December 19, 2005, from: www.w3.org/TR/201/REC-xmlschema-0-

20010502, downloaded pages 1-67, [hereinafter "XML Schema"], and further in view of

W3C, "XML Schema Requirements, W3C Note 15 February 1999," last downloaded by

the Examiner on December 19, 2005, from: www.w3.org/TR/NOTE-xml-schema-req,

downloaded pages 1-5, [hereinafter "XML Requirements"].

Regarding **dependent claim 2, as amended**, Ayers in view of XML Schema and further

in view of XML Requirements teaches:

> *The method of Claim 1, further comprising determining whether the field is*
>
> *one of a complex field or a simple field.*

(See, Ayers, page 3, screenshot and the last two full paragraphs, wherein the

screenshot teaches both simple and complex fields, with simple and complex elements,

and the code example teaches that the simple field is mapped and stored. Ayers

teaches the screenshot with the text to be saved, and Ayers teaches code generated in

XML from saving the text in AbiWord. Ayers does not expressly teach to determine

whether the field is one of a complex field and a simple field.

XML Schema teaches that simple and complex data elements are defined as

part of the XML language. See, XML Schema, downloaded page 7.

Ayers and SML Schema are analogous art because they are from the same field

of endeavor of creating and manipulating electronic documents. At the time the

invention was made, it would have been obvious to one of ordinary skill in the art to

differentiate between a simple element and a complex element, and to use such

differentiation to determine how to render a word document. It is at least obvious, and possibly inherent, from the screenshot and the code example shown in Ayers that AbiWord determines whether the data is complex or simple.

The suggestion or motivation for combining the teachings of Ayers, that a word processor program could convert and store documents in XML, with the teaching of XML Schema, including that document types are distinguished as simple or complex, is expressly stated in XML Requirements, stating: "The following usage scenarios describe XML applications that should benefit from XML schemas. * * * 4. Traditional document authoring/editing governed by schema constraints." See, XML Requirements, page 3.

The combination of the teachings of Ayers to convert and store word processing documents in XML combined with the teachings of SML Schema that data may be one of two types, simple or complex, would be an invention whereby a document mapped to XML would also distinguish and map simple and complex elements, and that such could be stored in memory.)

Regarding **dependent claim 3**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

> *The method of Claim 2, wherein an instruction portion of the field comprises at least one of richly formatted text and embedded additional fields when the field is a complex field.*

(The rejection of claim 2 is incorporated herein by this reference. See also, Ayers, page 2, third full paragraph, teaching that AbiWord can be saved in RTF format.)

Regarding **dependent claim 4**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

> *The method of Claim 2, wherein an instruction portion of the field excludes richly formatted text and embedded additional fields when the field is a simple field.*

(The rejection of claim 2 is incorporated herein by this reference.

It is noted that the term "richly formatted text" is not defined in the application. It is believed that applicants intended to refer to Microsoft Corporation's Rich Text Format (RTF) standard, and this definition will be used for the remainder of this Office Action.

It is inherent in the XML schema that simple types cannot have element content and cannot carry attributes, and it is therefore inherent in AbiWord, as taught by Ayers to save documents in XML. See in support of this inherent property, XML Schema, downloaded page 7.)

Regarding **dependent claim 5**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

> *The method of Claim 2, further comprising representing the field with a fldSimple element when the field is determined to be a simple field.*

(The rejection of claim 2 is incorporated herein by this reference.

It is noted that the term "fldSimple" is not expressly defined in the application. It is believed that the applicants created the term "fldSimple" as and element type.

It is further noted that the ability to create an element type is inherent in the XML language and is therefore inherent in AbiWord, which is taught by Ayers to save documents in XML.)

Regarding **dependent claim 6**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

> *The method of Claim 2, further comprising representing the field with at least one of a fldChar element and instrText element when the field is determined to be a complex field.*

(The rejection of claim 2 is incorporated herein by this reference.

It is noted that the ability to create an element type is inherent in the XML language and is therefore inherent in AbiWord, which is taught by Ayers to save documents in XML.)

Regarding **independent claim 11, as amended**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

> *A computer-readable medium for representing fields in a markup language document, comprising:*
>
> *inputting an application document that has been generated by an application that uses a file format that is specific to the application;*

(The specification discloses that the invention "provides a method to represent an application's native field structures in markup language (ML) such as XML." See, Disclosure, page 6, lines 12-13. Ayers teaches: "The most significant difference between AbiWord and nearly every other word processor available is the nature of the native file format. An *.abw file is written in XML and thus is also in ASCII format; the files can be read by any text editor." See, Ayers, page 2, fourth paragraph. Therefore, Ayers teaches the limitation of "a document that has been generated by an application that uses a file format that is specific to the application," and more specifically, teaches a native file format in a markup language, specifically XML.)

> *determining properties relating to one or more fields used within the application document;*

(It is noted that the term "properties" is not specifically defined in the specification. The disclosure associates the term in a general sense with simple and complex fields, as in the following: "the properties of the complex field (when the field is a complex field) are mapped into elements, attributes, and values of the ML file." Disclosure, page 18, lines 9-10. "Attributes" are also noted as one of the properties elsewhere in the disclosure. See, disclosure, page 3, line 26. Further evidence that the term refers to the general appearance of the document is taken from the disclosure, stating: "the fields and the properties associated with the fields may change from page to page, section to section, chapter to chapter and the like." Disclosure, page 18, lines 22-23. Based on the above analysis, it is believed that the applicants intended the term "properties" to be used in the general sense of the appearance of the text, and such definition will be used for the

remainder of this Office Action. In support of the above definition of "properties" as

known to one of ordinary skill in the art at the time of the invention, see, Harold, Elliotte

Rusty, "XML Bible," IDG Books Worldwide, 1999, pages 369-388.)

determining whether the field is one of a complex field and a simple field;

(See, Ayers, page 3, screenshot and the last two full paragraphs, wherein the

screenshot teaches both simple and complex fields, with simple and complex elements,

and the code example teaches that the simple field is mapped and stored. Ayers

teaches the screenshot with the text to be saved, and Ayers teaches code generated in

XML from saving the text in AbiWord. Ayers does not expressly teach to determine

whether the field is one of a complex field and a simple field.

XML Schema teaches that simple and complex data elements are defined as

part of the XML language. See, XML Schema, downloaded page 7.

Ayers and SML Schema are analogous art because they are from the same field

of endeavor of creating and manipulating electronic documents. At the time the

invention was made, it would have been obvious to one of ordinary skill in the art to

differentiate between a simple element and a complex element, and to use such

differentiation to determine how to render a word document. It is at least obvious, and

possibly inherent, from the screenshot and the code example shown in Ayers that

AbiWord determines whether the data is complex or simple.

The suggestion or motivation for combining the teachings of Ayers, that a word

processor program could convert and store documents in XML, with the teaching of

XML Schema, including that document types are distinguished as simple or complex, is

expressly stated in XML Requirements, stating: "The following usage scenarios describe XML applications that should benefit from XML schemas. * * * 4. Traditional document authoring/editing governed by schema constraints." See, XML Requirements, page 3.

The combination of the teachings of Ayers to convert and store word processing documents in XML combined with the teachings of SML Schema that data may be one of two types, simple or complex, would be an invention whereby a document mapped to XML would also distinguish and map simple and complex elements, and that such could be stored in memory.)

> *writing the properties into at least one of a markup language element, an*
>
> *attribute, and a value; and*

(See, Ayers, page 3, last two full paragraphs and citing code example from the screenshot and showing font and line properties and values. Note that Ayers teaches that the code is the markup language mapped from the properties of the document field displayed in the screenshot.)

> *storing the properties in the markup language document such that the*
>
> *fields of the application document are substantially maintained when the markup*
>
> *language document is parsed by an application that is different from the*
>
> *application used to generate the application document.*

(See, Ayers, page 3, last two full paragraphs and citing code example from the screenshot. Note that Ayers teaches saving the document shown in the screenshot, which creates the saved code that is saved in memory. Ayers also teaches: "The most

significant difference between AbiWord and nearly every other word processor available

is the nature of the native file format. An *.abw file is written in XML and thus is also in

ASCII format; **the files can be read by any text editor**." See, Ayers, page 2, fourth

paragraph, emphasis added.)

Regarding **dependent claim 12**, claim 12 incorporates substantially similar subject

matter as claimed in claim 9, and is rejected along the same rationale.

Regarding **dependent claim 13**, claim 13 incorporates substantially similar subject

matter as claimed in claim 10, and is rejected along the same rationale.

Regarding **dependent claim 14**, claim 14 incorporates substantially similar subject

matter as claimed in claim 3, and is rejected along the same rationale.

Regarding **dependent claim 15**, claim 15 incorporates substantially similar subject

matter as claimed in claim 4, and is rejected along the same rationale.

Regarding **dependent claim 16**, claim 16 incorporates substantially similar subject

matter as claimed in claim 5, and is rejected along the same rationale.

Regarding **dependent claim 17**, claim 17 incorporates substantially similar subject

matter as claimed in claim 6, and is rejected along the same rationale.

Regarding **dependent claim 18**, claim 18 incorporates substantially similar subject

matter as claimed in claim 7, and is rejected along the same rationale.

Regarding **independent claim 19**, Ayers in view of XML Schema and further in view of

XML Requirements teaches:

A system for representing fields in a markup language document,

comprising:

an application that is configured to:

input an application document that has been generated by an

application that uses a file format that is specific to the application;

determine properties relating to a field included in at least one

section of an application document;

determine whether the field is one of a complex field and a simple

field;

map the properties into at least one of a markup language

element, an attribute, and a value; and

store the properties in the markup language document; and

a validation engine configured to validate the markup language document.

(Claim 19 incorporates substantially similar subject matter as claimed in claim 11, and in

further view of the following, is rejected along the same rationale.

Validation of an XML document is expressly taught in XML Schema, which

states: "An instance document may be processed against a schema to verify whether

the rules specified in the schema are honored in the instance. Typically, such

processing actually does two things, (1) it checks for conformance to the rules, a

process called schema validation . . .." See, XML Schema, downloaded page 59. See

also, XML Schema, downloaded page 60, expressly teaching validation of simple and

complex element types. It would have been obvious to one of ordinary skill in the art at

the time of the invention to validate an instance document in order to confirm that it

follows the rules of the schema.

The specification discloses that the invention "provides a method to represent an

application's native field structures in markup language (ML) such as XML." See,

Disclosure, page 6, lines 12-13. Ayers teaches: "The most significant difference

between AbiWord and nearly every other word processor available is the nature of the

native file format. An *.abw file is written in XML and thus is also in ASCII format; the

files can be read by any text editor." See, Ayers, page 2, fourth paragraph. Therefore,

Ayers teaches the limitation of "a document that has been generated by an application

that uses a file format that is specific to the application," and more specifically, teaches

a native file format in a markup language, specifically XML.)

Regarding **dependent claim 20**, claim 20 incorporates substantially similar subject

matter as claimed in claim 9, and is rejected along the same rationale.

Regarding **dependent claim 21**, claim 21 incorporates substantially similar subject

matter as claimed in claim 3, and is rejected along the same rationale.

Regarding **dependent claim 22**, claim 22 incorporates substantially similar subject

matter as claimed in claim 4, and is rejected along the same rationale.

Regarding **dependent claim 23**, claim 23 incorporates substantially similar subject

matter as claimed in claim 10, and is rejected along the same rationale.

19.   It is noted that any citations to specific, pages, columns, lines, or figures in the

prior art references and any interpretation of the references should not be considered to

be limiting in any way. A reference is relevant for all it contains and may be relied upon

for all that it would have reasonably suggested to one having ordinary skill in the art.

See, MPEP 2123.

Response to Arguments

Applicants' arguments filed March 29, 2006 have been fully considered, but they

are not persuasive.

Regarding objection to the drawings:

First: Applicant argues that the specification description of the plural

"instructions 320" is a clear identification of the substance of the singular element

identified as "320" in figure 3. See, Amendment, page 8.

The Examiner disagrees.

Initially, it is noted that the specification indicates more than one instruction, yet there is only one figure. Thus, the definition of element 320 is not clear from the specification.

In addition, element 320 includes 🔲 an arrow. The meaning of the arrow indication is not clear. The rules for drawings identify three uses for an arrow: 1) to indicate the entire section to which it points, 2) to indicate the surface shown by the line looking along the direction of the arrow, or 3) to show the direction of movement. See, 37 CFR 1.84(r). None of the above situations apply to element 320.

Therefore, element 320, as indicated by an arrow, is not clear and must be amended.

Second: Applicant argues that reference characters 440 are made clear by reference to the specification that they are "instructions." See, Amendment, page 8.

The Examiner disagrees.

The first "440" and the second "440" are different instructions. Looking at the drawings, it is not clear why there are two different data elements, both identified as "440." Merely declaring that they are both "instructions" does not sufficiently make the entries clear. It is suggested that the Applicant indicate the similarity of the types of data shown as similar in type but different in expression by using reference characters such as "440A" and "440B" or "440a" and "440b", or different numbers.

Third: Applicant argues that reference characters 310, 330, 410, 420, 430, and 450, "clearly point to lexical text elements delimited by whitespace, metabrackets, and other non-character symbols that, for example, a front-end lexical analyzer of a compiler is capable of parsing." See, Amendment, page 8.

The Examiner disagrees that the figures identified by the reference characters is clear.

Applicant may be correct that the characters point to computer language which is "lexical text elements delimited by whitespace, metabrackets, and other non-character symbols that, for example, a front-end lexical analyzer of a compiler is capable of parsing." The Examiner disagrees that the characters "clearly point" to such complex meanings from the figures. For example, the figures also point to the letter "D," the symbol ">," the word "fldChar," the number "0," and many other reasonable interpretations of the identity of the drawing elements associated with the reference characters.

Regarding the objection to the Abstract:

Applicants traverse the objection arguing that the Abstract "does not require that which is new in the art to which the invention pertains" to be included in a concise statement. Quoting from the form paragraph which reminds Applicant of the proper content of the abstract of the disclosure, Applicant argues further that the use of the word "should include," in the form paragraph, makes merely optional the disclosure of

what is "new in the art to which the invention pertains."

The Examiner disagrees.

As stated in the Rules: "The purpose of the abstract is to enable the United States Patent and Trademark Office and the public generally to determine quickly from a cursory inspection the nature and gist of the technical disclosure." See, 37 CFR 1.72.

In the case of the present disclosure, the Applicant describes properties of markup languages, not properties of the invention. The abstract, therefore, is not sufficient to enable the Patent Office and the public to quickly, from a cursory inspection, determine the nature and gist of the technical disclosure, the invention. Accordingly, the abstract in its present form does not meet the purpose of the abstract, and is therefore objected to.

Objection to the Specification:

Applicant argues that it has properly submitted the computer code from the specification in a compact disc format.

The Examiner disagrees.

The Examiner believes the Applicant made a good faith attempt to properly submit the computer code on compact disc, but the disc is not acceptable in the present format. The Applicant has submitted a compact disc. However, the code on the compact disc submitted by the Applicants does not conform to the standards set forth in 37 CFR 1.96(c),as required in the Non-Final Office Action. Specifically, the data

submitted on the compact disc is not in American Standard Code for Information

Interchange (ASCII) format. See, 37 CFR 1.96(c)(3)(i).

Regarding rejections of claim 1 under 35 U.S.C. 102(b):

First: Applicants argue that the Examiner is incorrect in his derived definition of

the term "properties" as being "used in the general sense of the appearance of the text,

irrespective of the font" See, Non-Final Office Action, page 8. See, Amendment, page

10. In support, Applicant's argue that the term "property" includes an "attribute." See,

Amendment, page 10, citing disclosure, page 3, line 26.

The Examiner disagrees.

It is noted that the term "properties" is not specifically defined in the specification.

The disclosure associates the term in a general sense with simple and complex fields,

as in the following: "the properties of the complex field (when the field is a complex

field) are mapped into elements, attributes, and values of the ML file." Disclosure, page

18, lines 9-10. "Attributes" are also noted as one of the properties elsewhere in the

disclosure. See, disclosure, page 3, line 26. The disclosure also defines that an

element may have no attribute associated with it. See, disclosure, page 3, lines 27-28.

Further evidence that the term refers to the general appearance of the document is

taken from the disclosure, stating: "the fields and the properties associated with the

fields may change from page to page, section to section, chapter to chapter and the

like." Disclosure, page 18, lines 22-23. Based on the above analysis, it is believed that

the applicants intended the term "properties" to be used in the general sense of the

appearance of the text, and such definition will be used for the remainder of this Office

Action. In support of the above definition of "properties" as known to one of ordinary

skill in the art at the time of the invention, see, Harold, Elliotte Rusty, "XML Bible," IDG

Books Worldwide, 1999, pages 369-388. Although the Examiner agrees that an

"attribute" is within the "properties" claimed, the broadest reasonable interpretation of

the claim language is consistent with a reading of "properties" being used in the general

sense of the appearance of the text.

Second: Applicants argue that Ayers fails to teach of suggest inputting an

application document that has been generated by an application that uses a file format

that is specific to the application. See, Amendment, pages 10-11.

The Examiner disagrees.

The limitation of "a document that has been generated by an application that

uses a file format that is specific to the application" was only added in the recent

amendment.

The specification discloses that the invention "provides a method to represent an

application's native field structures in markup language (ML) such as XML." See,

Disclosure, page 6, lines 12-13. Ayers teaches: "The most significant difference

between AbiWord and nearly every other word processor available is the nature of the

native file format. An *.abw file is written in XML and thus is also in ASCII format; the

files can be read by any text editor." See, Ayers, page 2, fourth paragraph. Therefore,

Ayers teaches the limitation of "a document that has been generated by an application

that uses a file format that is specific to the application," and more specifically, teaches

a native file format in a markup language, specifically XML.

Comment on Arguments Regarding Claims 2 and 7:

The Amendment, page 11, first full paragraph, facially argues limitations of both

claims 2 and 7, but does not distinguish between the claims. However, specifically as to

claim 7, there is no argument, merely a first sentence that discusses what Ayers

teaches, and a final sentence that states that the claim is believed to be allowable. The

argument in the middle of the paragraph is directed to Claim 2. The Examiner cannot

discern whether the citations are a typo, or whether both claims were intended to be

covered by the arguments addressed to claim 2. Accordingly, the Examiner will

respond as follows assuming that the argument addressed to claim 2 was intended to

apply to both claims 2 and 7.

Regarding rejection of claim 2, as amended, under 35 U.S.C. 102(b):

First: Applicants argue that claim 2 is allowable at least for the reasons argued

for claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claim 1, above, in addition to the reasons

cited in rejection of claim 2, above, claim 2 is not in a condition of allowability.

Second: Applicants argue that Ayers fails to teach or suggest determining an

additional field from an application document using a file format that is specific to the

application. Further, Applicant argues that this distinction is significant because Ayers

does not determine filed properties from an application document using a file format that is specific to the application. See, Amendment, page 11.

The Examiner disagrees.

The limitation of "determining an additional field from an application document using a file format that is specific to the application" is not specified in claim 2, as currently amended.

Regarding rejection of claim 7 under 35 U.S.C. 102(b):

Applicants argue that Ayers fails to teach or suggest determining an additional field from an application document using a file format that is specific to the application. Further, Applicant argues that this distinction is significant because Ayers does not determine filed properties from an application document using a file format that is specific to the application. See, Amendment, page 11.

The Examiner disagrees.

The limitation of "determining an additional field from an application document using a file format that is specific to the application" is not specified in claim 7.

Regarding rejection of claim 9 under 35 U.S.C. 102(b):

First: Applicants argue that claim 9 is allowable at least for the reasons argued for claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claim 1, above, in addition to the reasons

cited in rejection of claim 9, above, claim 9 is not in a condition of allowability.

Second: Applicants argue that Ayers fails to teach or suggest that properties of

the fields stored in the markup language document are understood by an application

that understands the markup language when the field is not native to the application.

See, Amendment, page 11.

The Examiner disagrees.

Ayers teaches fields stored in XML that are understood by an application that

understands the markup language when the field is not native to the application. See,

Ayers, page 2, fourth paragraph teaching, as follows: "AbiWord can also save in the

HTML and RTF formats, both of which are accessible with word processors such as

MS-Word and WordPerfect." Therefore, AbiWord's fields in XML are understood by

markup languages not native to AbiWord, such as HTML.

Regarding rejection of claim 10 under 35 U.S.C. 102(b):

First: Applicants argue that claim 10 is allowable at least for the reasons argued

for claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claim 1, above, in addition to the reasons

cited in rejection of claim 10, above, claim 10 is not in a condition of allowability.

Regarding rejection of claim 2 under 35 U.S.C. 103(a):

Applicants argue that claim 2 is allowable at least for the reasons argued for

claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claim 1, above, in addition to the reasons

cited in rejection of claim 2, above, claim 2 is not in a condition of allowability.

Regarding rejection of claims 3-6 under 35 U.S.C. 103(a):

Applicants argue that claims 3-6 are allowable at least for the reasons argued for

claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claim 2, above, in addition to the reasons

cited in rejection of claims 3-6, above, claims 3-6 are not in a condition of allowability.

Regarding rejection of claim 8:

First: Applicants argue that claim 8 is allowable at least for the reasons argued

for claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claim 1, above, in addition to the reasons

cited in rejection of claim 8, above, claim 8 is not in a condition of allowability.

Second: Applicants argue that the references do not teach or suggest

"processing further fields when the properties associated with all fields have not been

stored in the markup language document." Applicant further argues that the limitation is

significant because "the cited references do not determine field properties from an

application document using a file format that is specific to the application and

subsequently process all of the fields for storage in a markup language format." See,

Amendment, page 13.

The Examiner disagrees.

It is noted that claim 1 specifies storing "the filed" which is "a field." See, claim 1,

lines, 5 and 9. Claim 8 further limits claim 1 by specifying a repetition of the field

storage process until "all fields" have been stored. Claim 8 merely specifies completing

the field storage process, which is the equivalent of storing the document.

Ayers teaches the storage of the document, which teaches the storage of all

fields in the document. See, Ayers, page 3, last two full paragraphs and citing code

example from the screenshot. Note that Ayers teaches saving the document shown in

the screenshot, which creates the saved code that is saved in memory.

Regarding rejection of claims 11-23 under 35 U.S.C. 103(a):

Applicants argue that claims 11-23 are allowable at least for the reasons argued

for claim 1.

The Examiner disagrees.

For the reasons cited in rejection of claims 1, 9, 10, 3, 4,5, 6, 7, 11, 9, 3, 4, and

10, respectively, above, in addition to the any additional reasons cited in rejection of

claims 11-23, above, claims 11-23 are not in a condition of allowability.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS for the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael K. Botts whose telephone number is 571-272-5533. The examiner can normally be reached on Monday through Friday 8:00-4:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system.   Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

MKB/mkb

HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100